

### REMARKS

Claims 30, 31, 34, 35, 38, and 39 have been allowed. Claims 5-7, 14-16, and 23-25 have been indicated as allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claims 5, 14 and 23 have amended to correct dependencies but still include at least the same limitations as the previously pending claims, therefore these claims should still be considered allowable if rewritten in independent form.

#### Claim Rejections – 35 U.S.C. § 103

Claims 1, 2, 4, 8-11, 13, 17-20, 22, 26-29, 32, 33, 36, and 37 were rejected under 35 U.S.C. §103(a) as allegedly being obvious in view of Ruf (U.S. Patent No. 6,077,313) in view of Bodin et al. (“A User Level Program Transformation Tool”, hereinafter “Bodin”). Applicant respectfully traverses this rejection with respect to the claims in issue.

#### Independent claims 1, 2, 10, 11

The Examiner states that “Ruf does not explicitly disclose the claimed an application program based on a script of a script-driven software tool, comprising automatically analyzing the script, and where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system; parsing the script into statement. However, Bodin discloses the claimed a transformation script system, (see page 180, cot. 2, paragraph 5), further in see page 155, cot. 1, paragraphs 1 -3, Bodin discloses a pattern that describes the code fragments to match before applying the transformation script. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the combined teachings of Ruf and Bodin with an application program based on a script of a script-driven software tool, comprising automatically analyzing the script, and where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system; parsing the script into statement” (pages 2 – 3 of the Office Action).

However, even if, for the sake of argument, the Examiner's characterization of Bodin's teachings were correct and there were sufficient motivation in the references or in the knowledge of one of ordinary skill in the art to combine and modify the teachings of Ruf and Bodin as the Examiner suggests (which we do not concede), that combination would still not disclose all of the limitations of these claims.

For example, claim 1 requires "parsing the script into statements comprising at least processing steps and dataset definitions." The Examiner states that "Ruf does not explicitly disclose ... parsing the script into statement," but that "It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the combined teachings of Ruf and Bodin with an application program based on a script of a script-driven software tool, comprising automatically analyzing the script, and where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system; parsing the script into statement" (page 3 of the Office Action). However, neither Ruf nor Bodin teach or suggest "parsing the script into statements comprising at least processing steps and dataset definitions."

Claim 1 also requires "constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results." The Examiner identifies col. 3, lines 16-22 of Ruf as disclosing this limitation. Instead, this portion of Ruf describes the following:

For dataflow analysis dependencies defined by a dependence relation, the dependence relation may be transformed by merging mutually dependent types in the dependence relation into type representatives such that the transformed dependence relation comprises type representatives each representing one or more types. A dependence relation represented by a dependence graph of nodes with each node representing one of the types may be transformed by collapsing each of one or more strongly-connected components of nodes representing mutually dependent types into a single node such that each node of the transformed dependence graph represents a type representative. (col. 3, lines 11 – 22)

Thus, Ruf describes a "dependence graph of nodes with each node representing one of the types," and not "a serial dataflow graph from the parsed statements, the serial dataflow graph

having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results.” The “types” in Ruf are described as follows:

The partitioning is performed based on a dependence relation over types representing run-time and therefore analysis-time values for the relevant program quantities in a type relation between program quantities and types for the program. A type analysis computes the type relation from user-declared types in the program or with a non-standard type inference analysis in accordance with the dataflow analysis to be performed. A dependence analysis determines the dependence relation among types of the computed type relation, and therefore among the analysis-time values of relevant program quantities, in accordance with the dataflow analysis to be performed. (col. 7, lines 14 – 25)

Claim 1 also requires “constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system.” The Examiner identifies col. 2, lines 48-52 of Ruf as disclosing “constructing a parallel dataflow graph from the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system” (page 5 of the Office Action). This portion of Ruf does not appear to be relevant; however, it appears that the Examiner meant to cite col. 1, lines 47 – 53 which describes the following:

Other methods to reduce the costs of performing dataflow analyses target factor (2) or both factors (2) and (3) by partitioning the analysis into phases, each of which models only a subset of the program quantities and/or points. Partitioning the dataflow analysis into phases helps reduce memory space costs as some dataflow analyses or optimizations may be performed on a per-phase basis. The storage used by the analysis for each phase may therefore be reclaimed for use by subsequent phases.

Thus, Ruf merely describes that dataflow analysis can be partitioned into phases, and that “more than one individual phase may be performed simultaneously in parallel” (col. 2, lines 5 – 7), but not that a parallel dataflow graph is constructed “from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system,” as required by claim 1.

These arguments also apply to claims 2, 10, 11, 19, and 20.

Independent claims 28, 29, 32, 33, 36, and 37

Ruf does not teach or suggest at least "constructing a parallel dataflow graph from the serial dataflow graph." As in the argument above, the portion of Ruf that the Examiner cites does not teach or suggest this limitation (and neither does any other portion of Ruf). Thus, even if, for the sake of argument, the Examiner's characterization of Bodin's teachings were correct and there were sufficient motivation in the references or in the knowledge of one of ordinary skill in the art to combine and modify the teachings of Ruf and Bodin as the Examiner suggests (which we do not concede), that combination would still not disclose all of the limitations of these claims.

Dependent claims 4, 8, 9, 13, 17, 18, 22, 26, and 27

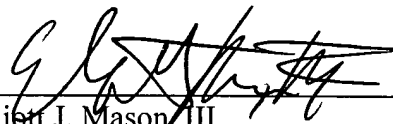
These claims are patentable for at least the same reasons as the claims from which they depend.

Accordingly, Applicant submits that claims 1, 2, 4, 8-11, 13, 17-20, 22, 26-29, 32, 33, 36, and 37 are allowable.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 3-28-05

  
\_\_\_\_\_  
Elliott J. Mason, III  
Reg. No. 56,569

Fish & Richardson P.C.  
225 Franklin Street  
Boston, MA 02110-2804  
Telephone: (617) 542-5070  
Facsimile: (617) 542-8906